

manuel89

Programmes du manuel utilisateur

Calculatrices : 89 92+ V200

Niveaux : 1ere S Tale Tale S

Descriptif : Groupe de programmes - Programmes du manuel utilisateur de la TI-89

Mots-clefs : manuel

Date de dernière révision : Septembre 2002

Présentation :

Il s'agit de l'ensemble des exemples de programmes du manuel utilisateur de la TI-89 :

- *def* (accompagné de *def1*, *def2* et *defset*) : recherche l'ensemble de définition d'une fonction

- *ineq* : résout des inéquations

- *kbdprgm3* : exemple de programme s'appelant par un raccourci-clavier (qui sert ici à créer un menu personnalisé).

Mode d'emploi :

Voici comment utiliser ces programmes :

- pour déterminer l'ensemble de définition de la fonction définie par $f(x)=expr$, il suffit de taper *def (expr)*. Utiliser *defset* de la même manière peut parfois être intéressant pour simplifier le résultat.

- pour résoudre une inéquation *ine*, il suffit de taper *ineq (ine)*.

- pour appeler *kbdprgm3*, il suffit d'appuyer sur [◊] puis 3. Vous pouvez naturellement taper *kbdprgm3()*.

Sources :

Def (ex)

Func

Local s,l,i,n

part(ex)→n

If n=0 Then

 true

Else

when(n=1,def1(ex),def2(ex))

EndIf

EndFunc

Def1 (ex)

Func

Local l,s,t

```

part(ex,0)→t
part(ex,1)→s

```

```

If t="ln" Then
  s>0→1
ElseIf t="√" Then
  s≥0→1
ElseIf t="-" Then
  true→1
Else
  df(ex)→1
EndIf
1 and def(s)
EndFunc

```

```

Def2 (ex)
Func
Local 1,t,s1,s2
part(ex,0)→t
part(ex,1)→s1
part(ex,2)→s2
If t="/" Then
  s2≠0→1
ElseIf t="^" Then
  when(int(s2)=s2,true,s1≥0,s1>0)→1
Else
  true→1
EndIf
def(s1) and def(s2) and 1→1
EndFunc

```

```

Defset (ex,x)
Func
Local eq
def(ex)→eq
not eq→eq
solve(eq,x)→eq
not eq
EndFunc

```

```

Ineq (ex)
Prgm
Local i,s,θt,d,lz1,lz2,lz,lf1,lf2,lf,la,sol,sp,den,num,test,ineg,poly2
Define
test(θex,θx,θt)=when(θt=1,approx(θex|x=θx)≤0,approx(θex|x=θx)<0)
Define ineq(θx,θy,θt)=when(θt=1,θx≤θy,θx<θy)
Define poly2(p)=when(d(p,x,3)=0,true,false,false)
setMode("exact/approx","AUTO")
DelVar x:ClrIO:Disp ex
If part(ex)=0:Stop
left(ex)-right(ex)→d:part(ex,0)→s
If s="<" Then:0→θt
ElseIf s=">" Then:~d→d:0→θt
ElseIf s="≤" Then:1→θt
ElseIf s="≥" Then:~d→d:1→θt

```

```

Else:Disp "Inequation invalide":Stop
EndIf

getDenom(d)→den:getNum(d)→num
Disp "Equation equivalente"
Pause ineg(num/den,0,θt)
If not poly2(den) or not poly2(num) Then
Disp "Inequation invalide"
Stop
EndIf
zeros(den,x)→lz1:
Disp "Etude num. et dén."
Disp den,"nul en ":Pause lz1
when(dim(lz1)>0,newList(dim(lz1)),{})→lf1
zeros(num,x)→lz2
Disp num,"nul en ":Pause lz2
when(dim(lz2)>0,newList(dim(lz2))+θt,{} )→lf2
augment(lz1,lz2)→lz:augment(lf1,lf2)→lf
If dim(lz)>0 Then
approx(lz)→la
SortA la,lz,lf
For i,1,dim(lz)-1
If la[i]=la[i+1] Then
lf[i]*lf[i+1]→lf[i]:lf[i]→lf[i+1]
EndIf
EndFor
test(d,lz[1]-10,θt)→s
when(s,ineg(x,lz[1],lf[1]),false)→sol
s→sp
For i,1,dim(lz)-1
If la[i+1]>la[i] Then
test(d,(lz[i]+lz[i+1])/2,θt)→s
If s
sol or ineg(lz[i],x,lf[i]) and ineg(x,lz[i+1],lf[i+1])→sol
If not s and not sp and lf[i]=1 and θt=1
sol or x=lz[i]→sol
s→sp
EndIf
EndFor
dim(lz)→i
test(d,lz[dim(lz)]+10,θt)→s
If s
sol or ineg(lz[i],x,lf[i])→sol
If not s and not sp and lf[i]=1 and θt=1
sol or x=lz[i]→sol
Else
test(d,0,θt)→sol
EndIf
Disp "Solutions :"
Pause when(sol,"S=R",{},sol)
DispHome
EndPrgm

Kbdprgm3 ( )
Prgm
Custom

```

```

Title  "Algebre"
Item  "factor("
Item  "expand("
Item  "comDenom("
Item  "getNum("
Item  "getDenom("
Title  "Equations"
Item  "solve("
Item  "zeros("
Item  " and "
Item  "left("
Item  "right("
Item  "solve( ,x)"
Item  "solve( and ,{x,y})"
Title  "Fonctions"
Item  "limit("
Item  "d("
Item  "f("
Item  "limit( ,x, )"
Item  "limit( ,x,-∞)"
Item  "limit( ,x,∞)"
Item  "limit( ,x, ,1)"
Item  "limit( ,x, ,-1)"
Item  "d( ,x)"
Item  "f( ,x)"
Title  "Trig&Cplx"
Item  "tCollect("
Item  "tExpand("
Item  "real(":
Item  "imag("
Item  "conj("
Item  "abs("
Item  "angle("
Item  "cSolve("
Title  "Divers"
Item  "ClrHome"
Item  "NewProb"
Item  "CustmOff"
EndCustm
CustmOn
EndPrgm

```